CASTEP user guide

Keith Refson
CCLRC Rutherford Appleton Laboratory
Chilton
Didcot
OXON OX12 0QX

September 7, 2004

1 Introduction

CASTEP is an *ab initio* density functional theory computer code using a plane-wave basis set and pseudpotentials. It is written in Fortran 90 and may be compiled for execution on either serial or parallel computers using the MPI communications interface.

CASTEP 2.2, 3.02 onwards is a complete re-implementation from scratch and shares no code with the older code of the same name, somewhat confusingly numbered CASTEP 4.2. Input and output file formats are not compatible with CASTEP 4.2.

This guide is still under construction and when finished will descript the capabilities, input files output formats and running instructions for CASTEP 3.02 and later version. It is not a reference on density functional theory or plane-wave methods and will presume that the reader is familiar with the theory and practice of plane-wave methods.

See references (7), (6), (2), (1),(8), (4), (3) for an introduction of the theory and methods used in CASTEP. There is now an excellent book on the subject by Richard Martin (5) and an accompanying website `http://electronicstructure.org`.

1.1 Capabilities of CASTEP

Castep can perform Total energy calculations on a supplied crystal structure ("single-point energy"), geometry optimization including simultaneous optimization of cell and internal co-ordinates, molecular dynamics in NVE and NVT ensembles and harmonic phonon frequencies and eigenvectors using density-functional perturbation therory.

- **Pseudopotentials**
  - Norm-conserving and Vanderbilt Ultrasoft pseudopotentials
  - Built-in pseudopotential generator (experimental).
  - Non-linear core charge correction.
- **Exchange and Correlation Functionals**
  - LDA - Local Density Approximation
  - PW91 - Perdew Wang '91 GGA
  - PBE - Perdew Burke Ernzerhof
  - RPBE - Revised Perdew Burke Ernzerhof
  - HF - Hartree-Fock
  - SHF - Screened Hartree-Fock

  - EXX - Exact exchange
  - SX - Screened exchange
  - ZERO - No exchange-correlation potential
  - HF-LDA -LDA with exchange contribution replaced by Hartree-Fock
  - SHF-LDA - -LDA with exchange contribution replaced by screened Hartree-Fock
  - EXX-LDA - -LDA with exchange contribution replaced by exact exchange
  - SX-LDA - LDA with exchange contribution replaced by screeened exchange

- **Electronic Minimization**
  - all-bands conjugate gradient minimizer for insulators
  - Density-mixing minimizer for metals
  - Ensemble DFT for metals
  - Double-grid for charge densities
  - Spin-polarized or unpolarized (paired electron) calculations

- **Geometry Optimization**
  - Robust BFGS optimizer with line search
  - Internal Co-ordinates optimizer for molecular systems
  - Combined atomic/cell optimization using augmented Hessian
  - Accurate variable-cell geometry optimization at fixed cutoff
  - Damped MD

- **Molecular Dynamics**
  - NVE and NVT ensemble MD simulation
  - Accurate variable-cell MD at fixed cutoff
  - Nose-Hoover chain thermostat

- **Transition-State Searching**

- **Density-Functional Linear Response**
  - Phonon Spectra and Dispersion Curves.
  - Dielectric permittivities (low frequency and optical)
  - Born effective charges
  - LO-TO splitting

2 Input files

To run, CASTEP requires two input files plus one or two additional pseudopotential files for every element included. The filenames of both input and output files begin with a common root, known as a "seed", making it easy to distinguish the files belonging to a particular run. To run it is merely necessary to execute the command (either interactively or from a shell-script from a batch queue)

`castep` *seed*

where we assume that the executable is named `castep` and is found somewhere in the shell path. The two input files are *seed*`.cell` which describes the simulation cell and its contents and *seed*`.param` which describes the type of run to be performed and any options which may be required. (In fact the *seed*`.param` may be omitted entirely in which case a single-point energy calculation with default parameters is performed.)

Data is input in a '`keyword <value>`' fashion, and input is independent of the ordering of the input file. The value may be preceded by '`=`', '`:`' or a blank space or tab.

The keywords are case insensitive (e.g. unitCell is equivalent to UnItceLL) and punctuation insensitive (unit.cell is equivalent to unit_cell is equivalent to unitcell). Punctuation characters are '`.`' and '`_`'.

There are 9 different types of entity which may be defined in the input files.

- String - A text string up to `file_maxpath` characters in length.
- Integer - An integer.
- Real - A real number.
- Physical - A physical quantity of one of the dimensions listed in Tables 4 and 5.
- Defined - An entry which is either present or absent in the file and takes no value.
- Boolean - A logical value.
- Real Vector - A three-vector of real values.
- Integer Vector - A three-vector of integer values.
- Block - A block of data to be interpreted by the calling subroutine.

2.1 Help facility

CASTEP may be invoked as

`castep -help`

or

`castep -help` *variable*

to give brief summary information on an input variable in either the *seed*`.cell` or *seed*`.param` files, or a class of variable. A useful aide memoire if you don't have a copy of this manual handy.

2.2 The cell file

The cell definition file is a free-format keyword driven text file. The file defines the initial cell for a calculation. The cell lattice vectors, ionic positions, sampling k-points in the Brillouin zone, cell symmetry, external pressure, constraints on motion of the ions or cell, the pseudopotentials representing each species and the mass of each species may be defined.

At the very least, the cell lattice vectors and ionic positions must be specified. Reasonable defaults, defined in tables 1 to 3 will be chosen for other variables not specified.

A list of keywords for the cell file is given in Tables 1 to 3.

The definitions of the keywords are given in more detail in the following subsections. For the purposes of the following definitions, all variables represented by R are defined to be real numbers, those represented by I are defined to be integers and those represented by C are characters.

2.2.1 Cell Lattice Vectors

The cell lattice vectors may be specified in Cartesian coordinates or in terms of the lattice vector magnitudes and the angles between them $(a, b, c, \alpha, \beta, \gamma)$. Only one of LATTICE_CART and LATTICE_ABC may occur in a cell definition file.

The definitions of these keywords are as follows:

`%BLOCK LATTICE_CART`
`[units]`

$$\begin{array}{lll} R_{1x} & R_{1y} & R_{1z} \\ R_{2x} & R_{2y} & R_{2z} \\ R_{3x} & R_{3y} & R_{3z} \end{array}$$

`%ENDBLOCK LATTICE_CART`

Keyword	Type	Default	Description
LATTICE_CART*	B	–	The cell lattice vectors in Cartesian coordinates.
LATTICE_ABC*	B	–	The cell lattice vectors specified in $a, b, c, \alpha, \beta, \gamma$ format.
POSITIONS_FRAC†	B	–	The positions of the ions in fraction coordinates with respect to the lattice vectors.
POSITIONS_ABS†	B	–	The positions of the ions in absolute coordinates.
POSITIONS_FRAC_PRODUCT†	B	–	The positions of the ions in the product structure for a transition state search, in fraction coordinates with respect to the lattice vectors.
POSITIONS_ABS_PRODUCT†	B	–	The positions of the ions in the product structure for a transition state search, in absolute coordinates.
POSITIONS_FRAC_INTERMEDIATE†	B	–	The positions of the ions in an estimate of the intermediate structure for a transition state search, in fraction coordinates with respect to the lattice vectors.
POSITIONS_ABS_INTERMEDIATE†	B	–	The positions of the ions in an estimate of the intermediate structure for a transition state search, in absolute coordinates.
IONIC_VELOCITIES	B	random	The velocities of the ions in Cartesian coordinates.
KPOINT_LIST‡	B	–	A list of k-points in the Brillouin zone with associated weights.
KPOINT_MP_GRID‡	W	–	The k-points defined as a Monkhorst-Pack grid by specifying the grid dimensions in each direction.
KPOINT_MP_SPACING‡	P	0.1 Å^{-1}	The k-points as a Monkhorst-Pack grid by specifying the maximum distance between k-points.
KPOINT_MP_OFFSET	V	0,0,0	The offset of the origin of the Monkhorst-Pack grid in fractional coordinates relative to the reciprocal lattice vectors.

Table 1: The keywords which may be specified in the cell definition file. Full details of their definitions may be found in Section **??**. Argument types are represented by, B for block data, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, V for a real vector and W for an integer vector.

* Only one of LATTICE_CART and LATTICE_ABC may be present in a cell file.

† Only one of POSITIONS_FRAC and POSITIONS_ABS may be present in a cell file.

‡ Only one of KPOINTS_LIST, KPOINTS_MP_GRID and KPOINTS_MP_SPACING may be present in a cell file.

Keyword	Type	Default	Description
BS_KPOINT_PATH*	B	–	A list of k-points in the Brillouin zone which defines the path along which a band-structure calculation will be performed.
BS_KPOINT_PATH_SPACING	P	0.1 $Å^{-1}$	Specifies the maximum spacing between k-points along the path for which a band structure calculation will be performed.
BS_KPOINT_LIST*	B	SCF k-points	A list of k-points at which a band-structure calculation will be performed.
OPTICS_KPOINT_LIST‡	B	–	A list of k-points in the Brillouin zone with associated weights for optical matrix element calculations.
OPTICS_KPOINT_MP_GRID‡	W	–	The k-points for optical matrix element calculations defined as a Monkhorst-Pack grid by specifying the grid dimensions in each direction.
OPTICS_KPOINT_MP_SPACING‡	P	0.1 $Å^{-1}$	The k-points for optical matrix element calculations as a Monkhorst-Pack grid by specifying the maximum distance between k-points.
OPTICS_KPOINT_MP_OFFSET	V	0,0,0	The offset of the origin of the Monkhorst-Pack grid for optical matrix element calculations in fractional coordinates relative to the reciprocal lattice vectors.
PHONON_KPOINT_PATH∘	B	–	A list of k-points in the Brillouin zone which defines the path along which a phonon calculation will be performed.
PHONON_KPOINT_PATH_SPACING	P	0.1 $Å^{-1}$	Specifies the maximum spacing between k-points along the path for which a phonon calculation will be performed.
PHONON_KPOINT_LIST∘	B	SCF k-points	A list of k-points at which a phonon calculation will be performed.
PHONON_GAMMA_DIRECTIONS	B	See text	The directions in which the gamma point will be approached for calculation of the LO/TO splitting.

Table 2: The keywords which may be specified in the cell definition file. Full details of their definitions may be found in Section **??**. Argument types are represented by, B for block data, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, V for a real vector and W for an integer vector.

* Only one of BS_KPOINT_PATH and BS_KPOINT_LIST may be present in a cell file.

‡ Only one of OPTICS_KPOINTS_LIST, OPTICS_KPOINTS_MP_GRID and OPTICS_KPOINTS_MP_SPACING may be present in a cell file.

∘ Only one of PHONON_KPOINT_PATH and PHONON_KPOINT_LIST may be present in a cell file.

$ Only one of SYMMETRY_GENERATE and SYMMETRY_OPS may be present in a cell file.

Keyword	Type	Default	Description
SYMMETRY_GENERATE$	D	no symmetry	If this is present, the highest symmtery group of the cell will found and the corresponding symmetry operations generated.
SYMMETRY_OPS$	B	no symmetry	The symmetry operations that apply to the cell.
SYMMETRY_TOL	P	0.01 Å	The tolerance within which symmetry will be enforced.
IONIC_CONSTRAINTS	B	no constraints	The constraints on the motion of ions during relaxation or MD.
FIX_ALL_IONS	L	`FALSE`	Constrain all ionic positions to remain fixed.
FIX_ALL_CELL	L	`FALSE`	Constrain all cell parameters to remain fixed.
FIX_COM	L	`TRUE`	Constrain the centre of mass of the ions to remain fixed.
CELL_CONSTRAINTS	B	no constraints	The constraints on changes in the cell shape during relaxation or MD.
SPECIES_MASS	B	atomic mass	The masses of the ionic species.
SPECIES_POT	B	see text	The names of the pseudopotentials associated with each species.
SPECIES_LCAO_STATES	B	see text	The number of angular momentum states to use in the LCAO basis set for this species when performing population analysis.
EXTERNAL_PRESSURE	B	no pressure	The external pressure tensor.

Table 3: The keywords which may be specified in the cell definition file. Full details of their definitions may be found in Section **??**. Argument types are represented by, B for block data, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, V for a real vector and W for an integer vector.

Here R_{1x} is the x-component of the first lattice vector, R_{2y} is the y-component of the second lattice vector etc.

`[units]` specifies the units in which the lattice vectors are defined. If not present, the default is Å.

`%BLOCK LATTICE_ABC`
`[units]`

$$\begin{array}{ccc} R_a & R_b & R_c \\ R_\alpha & R_\beta & R_\gamma \end{array}$$

`%ENDBLOCK LATTICE_ABC`

Here R_a is the value of a, R_γ is the value of γ etc. If the lattice is specified in this manner, the absolute orientation is arbitrary. In this case the orientation is defined by applying the following constraints:

- **a** lies along the x-axis
- **b** lies in the xy plane
- **c** forms a right-handed set with **a** and **b**

`[units]` specifies the units in which the lattice vector magnitudes are defined. If not present, the default is Å. Angles should be specified in degrees.

2.2.2 Ionic Positions

The ionic positions may be specified in fractional coordinates relative to the lattice vectors of the unit cell, or in absolute coordinates. Only one of POSITIONS_FRAC and POSITIONS_ABS may occur in a cell definition file.

`%BLOCK POSITIONS_FRAC`

$$\begin{array}{cccccc} CCC_1|I_1 & R_{1i} & R_{1j} & R_{1k} & [\text{MAGMOM} = mm_1] \\ CCC_2|I_2 & R_{2i} & R_{2j} & R_{2k} & [\text{MAGMOM} = mm_2] \\ \vdots & & & & \end{array}$$

`%ENDBLOCK END POSITIONS_FRAC`

The first entry on a line is the symbol or atomic number of the ionic species. The correct symbol will be looked up for the atomic species if the atomic number is specified. A symbol can have a maximum of three characters. The first alphabetical characters identify the element, from which default values for atomic mass etc.

The next three entries on a line in POSITIONS_FRAC are real numbers representing the position of the ion in fractions of the unit cell lattice vectors.

If the optional flag "MAGMOM" is present on a line, this sets the spin polarisation ($\frac{N^\uparrow - N^\downarrow}{N_{tot}}$) of the atom for initialisation of the spin density. If this flag is not present a non-spin polarised state will be assumed.

`%BLOCK POSITIONS_ABS`
`[units]`

$$\begin{array}{cccccc} CCC_1|I_1 & R_{1x} & R_{1y} & R_{1z} & [\text{MAGMOM} = mm_1] \\ CCC_2|I_2 & R_{2x} & R_{2y} & R_{2z} & [\text{MAGMOM} = mm_2] \\ \vdots & & & & \end{array}$$

`%ENDBLOCK POSITIONS_ABS`

The first entry on a line is the symbol or atomic number of the ionic species, as for POSITIONS_FRAC. The next three entries are real numbers representing the position of the ion in Cartesian coordinates.

`[units]` specifies the units in which the positions are defined. If not present, the default is Å.

The optional flag MAGMOM is defined above under `POSITIONS_FRAC`.

For transition state searches, structures for the product and intermediate geometry of the system may be input in blocks `POSITIONS_FRAC_PRODUCT` and `POSITIONS_FRAC_INTERMEDIATE` respectively, in fractional coordinates, or `POSITIONS_ABS_PRODUCT` and
`POSITIONS_ABS_INTERMEDIATE` respectively, in absolute coordinates. The format of these blocks is the same as `POSITIONS_FRAC` and `POSITIONS_ABS` as appropriate. The reactant structure will be taken from the main positions block.

2.2.3 Ionic Velocities

The initial ionic velocities may be specified in Cartesian coordinates in a cell definition file.
`%BLOCK IONIC_VELOCITIES`
`[units]`

$$\begin{array}{llll} CCC_1|I_1 & V_{1x} & V_{1y} & V_{1z} \\ CCC_2|I_2 & V_{2x} & V_{2y} & V_{2z} \\ \vdots & & & \end{array}$$

`%ENDBLOCK IONIC_VELOCITIES`

The first entry on a line is the symbol or atomic number of the ionic species. The correct symbol will be looked up for the atomic species if the atomic number is specified. A symbol can have a maximum of three characters. The next three entries are real numbers representing the velocity of the ion in Cartesian coordinates.

`[units]` specifies the units in which the positions are defined. If not present, the default is Å/ps.

If this keyword is not present and a molecular dynamics calculation is performed, the ionic velocities will be randomly initialised with the appropriate temperature.

2.2.4 Brillouin Zone Sampling K-points

(N.B. in the following section the keywords the prefixes `KPOINT_` and `KPOINTS_` are synonymous. `KPOINT_` is the preferred usage.)

The k-points at which the Brillouin zone is to be sampled during a self consistent calculation to find the electronic ground state may be defined either by specifying a list of k-points or a Monkhorst-Pack grid in terms of the dimensions of the k-point mesh or a minimum k-point density. The origin of the Monkhorst-Pack grid may be offset by a vector from the origin of the Brillouin zone.

If no k-points are specified, the default will be a Monkhorst-Pack grid with a maximum spacing of 0.1Å^{-1} and no offset of the origin.

The KPOINT_LIST, KPOINT_MP_GRID and KPOINT_MP_SPACING keywords are mutually exclusive. KPOINT_MP_OFFSET may be specified in combination with either
KPOINT_MP_GRID or KPOINT_MP_SPACING.
`%BLOCK KPOINT_LIST`

$$\begin{array}{llll} R_{1i} & R_{1j} & R_{1k} & R_{1w} \\ R_{2i} & R_{2j} & R_{2k} & R_{2w} \\ \vdots & & & \end{array}$$

`%ENDBLOCK KPOINT_LIST`

The first three entries on a line are the fractional positions of the k-point relative to the reciprocal space lattice vectors. The final entry on a line is the weight of the k-point relative to the others specified. The sum of the weights must be equal to 1.

`KPOINT_MP_GRID` $I_i \quad I_j \quad I_k$

This specifies the dimensions of the Monkhorst-Pack grid requested in the directions of the reciprocal space lattice vectors.

`KPOINT_MP_SPACING` R `[units]`

The single entry is the maximum distance between k-points on the Monkhorst-Pack grid. The dimensions of the grid will be chosen such that the maximum separation of k-points is less than this.

`[units]` specifies the units in which the k-point spacing is defined. If not present, the default is Å^{-1}.

`KPOINT_MP_OFFSET` $R_i \quad R_j \quad R_k$

This specifies the offset of the Monkhorst-Pack grid with respect to the origin of the Brillouin zone. The three entries are the offset in fractional coordinates relative to the reciprocal lattice vectors.

The k-point set for performing optical matrix element calculations can be specified in the same manner, using version of the keywords above with OPTICS_ prepended. The same restrictions regarding mutually exclusive keywords apply.

For a non-self-consistent band structure calculation, the k-points may be defined along a path through reciprocal space or a list of k-points.

%BLOCK BS_KPOINT_PATH

$$\begin{matrix} R_{1i} & R_{1j} & R_{1k} \\ R_{2i} & R_{2j} & R_{2k} \\ \vdots & & \end{matrix}$$

%ENDBLOCK BS_KPOINT_PATH

The three numbers on each line are the fractional positions of the k-point relative to the reciprocal space lattice vectors. The k-points define a continuous sequence of straight line segments, unless the keyword BREAK appears on a separate line within the sequence of k-points. In this case the continuous path will end at the k-point immediately preceding the BREAK keyword and resume at the k-point immediately following. The path will be open unless the first and last point in the list are identical. The maximum spacing of the points sampled along each line segment is defined by the keyword BS_KPOINT_PATH_SPACING (default value 0.1 Å^{-1}. If necessary, the actual spacing used may be smaller than this in order to ensure that the length of the line segment is an integer multiple of the spacing between points on that segment.

Alternatively, the k-point set for performing a band structure calculation can be specified in the same manner as the main k-point set, using version of the keywords above with BS_ prepended. The same restrictions regarding mutually exclusive keywords apply. In this case, the k-point weight in BS_KPOINT_LIST is optional. If ommitted, the weights for each k-point are assumed to be equal.

For a phonon spectum calculation, the k-points may be defined along a path through reciprocal space or a list of k-points, in the same manner as for a band structure calculation. The corresponding keywords are identical to those for the band structure specification with the initial BS_ replaced by PHONON_, e.g. PHONON_KPOINT_PATH,
PHONON_KPOINT_PATH_SPACING and PHONON_KPOINT_LIST. The same restrictions regarding mutually exclusive keywords apply.

The block keyword PHONON_GAMMA_DIRECTIONS specifies the directions in which the gamma point will be approached when calculating the non-analytic terms of the LO/TO splitting. Each line in this block will consist of a 3-vector specifying a direction in the basis of reciprocal lattice vectors. If this keyword is not present, the default will be a single vector determined as follows:

1. If the gamma point is $q_i = 0$ and there is an successor kpoint q_{i+1} in the list then it is q_{i+1}.
2. Otherwise if the gamma point is $q_i = 0$ and there is an predecessor kpoint q_{i-1} in the list then it is q_{i-1}.
3. Otherwise (i.e. a Gamma point only calculation) the a-axis of the reciprocal cell.

For backwards compatibility the kewords beginning BS_KPOINT_ are synonyms for BS_KPOINT_ and similarly those beginning PONON_KPOINT_ are synonymous with PHONON_KPOINT_.

2.2.5 Cell Symmetry

The symmetry of the cell is represented as a series of symmetry operations under which the unit cell is invariant. Each operation is represented as a 3×3 array.

If no symmetry is specified in the cell definition file, the default is for no symmetry to be applied.

%BLOCK SYMMETRY_OPS

$$\begin{matrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \\ T_1 & T_2 & T_3 \\ R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \\ T_1 & T_2 & T_3 \vdots \end{matrix}$$

%ENDBLOCK SYMMETRY_OPS

Each of the first three lines contains 3 entries representing a row of a 3×3 array. These represent one symmetry rotation. The three entries on the following line contain the translation associated with this rotation.

`SYMMETRY_GENERATE`

If this keyword is persent in the cell, the highest symmetry group that applies to the structure of the cell will be found and the corresponding symmetry operations generated.

`SYMMETRY_TOL` R `[units]`

This parameter is the tolerance within which symmetry will be considered to be satisfied. If an ion is found within this distance of its symmetric position, the symmetry will be considered to be satisfied.

`[units]` specifies the units in which the tolerance is defined. If not present, the default is Å.

2.2.6 Constraints

The movement of ions or the unit cell during a relaxation or molecular dynamics run may be constrained.

The constraints on the ionic motion may by specified as a set of linear constraints. Each constraint is specified as a series of coefficients a_{ijk} such that:

$$\sum_{k=1}^{\texttt{num_species}} \sum_{j=1}^{\texttt{num_ions_in_species(k)}} \sum_{i=1}^{3} a_{ijk}\ \texttt{ionic_positions(i,j,k)} = constant$$

The change in the shape of the unit cell may also be constrained using the keyword CELL_CONSTRAINTS.

The special case of constraining the centre of mass of the ions to remain fixed is supported by a logical keyword FIX_COM. Also all ionic positions or cell parameters may be fixed by specifying the keywords FIX_ALL_IONS or FIX_ALL_CELL to be `TRUE` respectively.

If no ionic or cell constraints are specified in the cell definition file, the default is to fix the centre of mass.

`%BLOCK IONIC_CONSTRAINTS`

$$\begin{array}{llllll} I_1 & CCC_{1s}|I_{1s} & I_{n1} & R_{1i} & R_{1j} & R_{1k} \\ I_2 & CCC_{2s}|I_{2s} & I_{n2} & R_{2i} & R_{2j} & R_{2k} \\ \vdots & & & & & \end{array}$$

`%ENDBLOCK IONIC_CONSTRAINTS`

The first element on each line is an integer specifying the number of the constraint being specified. The second entry is either the symbol or atomic number of the species of the ion to which this constraint applies. The third element is the number of the ion within the species. The ordering of the ions in a species is the order in which they appear in the POSITIONS_FRAC or POSITIONS_ABS block in the cell definition file. The final three numbers are real numbers representing the coefficients of the Cartesian coordinates of the ionic position in the constraint sum. All coefficients in the sum not explicitly specified will be zero.

On reading this data, the matrix of ionic constraints will be orthogonalised.

`%BLOCK CELL_CONSTRAINTS`

$$\begin{array}{lll} I_a & I_b & I_c \\ I_\alpha & I_\beta & I_\gamma \end{array}$$

`%ENDBLOCK CELL_CONSTRAINTS`

The first three entries relate to the magnitude of the three lattice vectors a, b, c and the second set of three entries to the angles α, β, γ.

If the value of the entry corresponding to a magnitude or angle is zero, this quantity will remain fixed. If two or three entries contain the same integer, the corresponding quantities will be constrained to have the same value. If a positive integer greater than 0 occurs in entries 1 through 3 the same integer cannot occur in entries 4 through 6 as this would imply that a vector length and angle must have the same value.

2.2.7 Species Characteristics

The mass of a species, the pseudopotential which represents the ion and the size of the LCAO basis set used for population anslsyis may be specified in the cell definition file.

`%BLOCK SPECIES_MASS`
`[units]`

$$\begin{array}{ll} CCC_1|I_1 & R_1 \\ CCC_2|I_2 & R_2 \\ \vdots & \end{array}$$

`%ENDBLOCK SPECIES_MASS`

`[units]` specifies the units in which the masses are defined. If not present, the default is atomic mass units.

The first entry on a line is the symbol or atomic number of the species. This must correspond with the species symbol or atomic number of the species in the POSITIONS_FRAC or POSITIONS_ABS block. The second entry on each line is the mass of that species. Not all species need appear in the SPECIES_MASS block, any not present will assume the default mass for that species. If the initial alphabetical symbol specified for a species is not a standard element symbol in the periodic table, the mass of the species must be specified.

`%BLOCK SPECIES_POT`

$$\begin{array}{ll} CCC_1|I_1 & < filename > \\ CCC_2|I_2 & < filename > \\ \vdots & \end{array}$$

`%ENDBLOCK SPECIES_POT`

The first entry on a line is the symbol or atomic number of the species. This must correspond with the species symbol or atomic number of the species in the POSITIONS_FRAC or POSITIONS_ABS block. The second entry on each line is the filename of the file containing the definition of the pseudopotential representing the ionic species. The file to which this refers may be a definition of the parameters of the pseudopotential which is to be generated at runtime, or an old-style pseudopotential definition containing the data for the pseudopotential.

Not all species need appear in the SPECIES_POT block. If a pseudopotential is not specified, the default pseudopotential parameters will be used to generate a pseudopotential for the element specified. If the initial alphabetical characters of a species label is not a standard element symbol in the periodic table, the potential for the species must be specified.

The charge on the ion for each species will be derived from the pseudopotential corresponding to that ion.

`%BLOCK SPECIES_LCAO_STATES`

$$\begin{array}{ll} CCC_1|I_1 & I_{B1} \\ CCC_2|I_2 & I_{B2} \\ \vdots & \end{array}$$

`%ENDBLOCK SPECIES_LCAO_STATES`

The first entry on a line is the symbol or atomic number of the species. This must correspond with the species symbol or atomic number of the species in the POSITIONS_FRAC or POSITIONS_ABS block. The second number is the number of angular momentum channels to use in the LCAO basis set for the species when performing population analysis. For example, to use the 2s and 2p states for C (The 1s state is a core state) this should be 2. By default, the number of states will be the appropriate number to complete the valence shell to the next noble gas. If shallow core states are excluded from a pseudopotential, the value of SPECIES_LCAO_STATES for that specied should be included in the cell file to ensure a meaningful basis set is used.

2.2.8 External Pressure

An external pressure may be applied to the unit cell by specifying a pressure tensor.

%BLOCK EXTERNAL_PRESSURE
[units]

$$\begin{matrix} R_{xx} & R_{xy} & R_{xz} \\ & R_{yy} & R_{yz} \\ & & R_{zz} \end{matrix}$$

%ENDBLOCK EXTERNAL_PRESSURE

[units] specifies the units in which the pressure is defined. If not present, the default is GPa.

Entry R_{xx} is the xx-component of the pressure, R_{xy} the xy-component etc.

The default is to apply no external pressure.

2.3 The parameters file

The parameters file is a free-format containing keyword-value pairs, one per line. Any of the characters #, ; or ! begins a comment and the rest of the line is ignored. Keyword lines take the form

keyword = *value*

and any of the chacacters :, =, space and TAB may be used to separate the keyword and value.

It is not necessary to input the value of all parameters used. Each has a sensible default and it is usually only necessary to specify a very small number explicitly. Parameters whose value is a physical quantity may be given in any of a variety of units.

Physical quantities may be followed by a unit (separated by a space). The quantity will be automatically converted to atomic units, as used internally, when the file is read. The units which are recognised by the code for each physical dimension are listed in Tables 4 and 5. If no units are provided in the input, the assumed units will be those shown in these tables.

The following are equivalent ways of defining a physical quantity:

```
AgeOfUniverse = 24.d0 s
Age_Of_Universe : 24.d0 S
AgeOfUniverse 24.d3 ms
```

The parameters file is not only read at the beginning of a run, but also periodically whenever a run is chackpointed – for example every few molecular dynamics iterations. This facility may be used to adjust certain of the parameters during the course of a run. In particular, adding the single line STOP will cause the run to write a .check file and halt at the next checkpoint.

2.4 character(len=file_maxpath) :: seedname

The seed used for generating filenames. This is determined from the command line arguments when running the executable and is set by subroutine parameters_read.

2.5 character :: comment

A comment string which may be used to label the output.

By default this string will be blank.

2.6 integer :: iprint

This indicates the level of verbosity of the output from 0, the bare minimum to 3, which corresponds to full debugging output.

The default value of this parameter is 1.

Unit	Abbreviation	Dimension	Identifier	Value (atomic units)
Bohr	Bohr	L	bohr,a0	1
Metre	m	L	m	$\frac{m_e c\alpha}{\hbar}$
Centimetre	cm	L	cm	$\frac{m_e c\alpha}{\hbar} \times 10^{-2}$
Nanometre	nm	L	nm	$\frac{m_e c\alpha}{\hbar} \times 10^{-9}$
Ångstrom*	A	L	ang	$\frac{m_e c\alpha}{\hbar} \times 10^{-10}$
Electron Mass	m_e	M	me	1
Unified Atomic Mass Unit*	amu	M	amu	$\frac{1}{N_A m_e} \times 10^{-3}$
Kilogram	kg	M	kg	$\frac{1}{m_e}$
Gram	g	M	g	$\frac{1}{m_e} \times 10^{-3}$
Atomic Unit of Time	aut	T	aut	1
Second	s	T	s	$\frac{c^2\alpha^2 m_e}{\hbar}$
Millisecond	ms	T	ms	$\frac{c^2\alpha^2 m_e}{\hbar} \times 10^{-3}$
Microsecond	mus	T	mus	$\frac{c^2\alpha^2 m_e}{\hbar} \times 10^{-6}$
Nanosecond	ns	T	ns	$\frac{c^2\alpha^2 m_e}{\hbar} \times 10^{-9}$
Picosecond*	ps	T	ps	$\frac{c^2\alpha^2 m_e}{\hbar} \times 10^{-12}$
Femtosecond	fs	T	fs	$\frac{c^2\alpha^2 m_e}{\hbar} \times 10^{-15}$
Elementary Charge*	e	C	e	1
Coulomb	Coulomb	C	c	$\frac{1}{e}$
Hartree	Ha	E	hartree,ha	1
Millihartree	mHa	E	mhartree	10^{-3}
Electron Volt*	eV	E	ev	$\frac{e}{\alpha^2 m_e c^2}$
Milli-Electron Volt	meV	E	mev	$\frac{e}{\alpha^2 m_e c^2} \times 10^{-3}$
Rydberg	Ry	E	ry	$\frac{1}{2}$
Millirydberg	mRy	E	mry	5×10^{-4}
kJ/mol	kJ/mol	E	kj/mol	$\frac{1\times10^3}{\alpha^2 m_e c^2 N_A}$
kcal/mol	kcal/mol	E	kcal/mol	$\frac{4.184\times10^3}{\alpha^2 m_e c^2 N_A}$
Joule	J	E	j	$\frac{1}{\alpha^2 m_e c^2}$
Erg	erg	E	erg	$\frac{1\times10^{-7}}{\alpha^2 m_e c^2}$
Hertz	Hz	E	hz	$\frac{h}{\alpha^2 m_e c^2}$
Megahertz	MHz	E	mhz	$\frac{h}{\alpha^2 m_e c^2} \times 10^6$
Gigahertz	GHz	E	ghz	$\frac{h}{\alpha^2 m_e c^2} \times 10^9$
Terahertz	THz	E	thz	$\frac{h}{\alpha^2 m_e c^2} \times 10^{12}$
Wavenumber	cm-1	E	cm-1	$\frac{hc}{\alpha^2 m_e c^2} \times 10^2$
Kelvin	K	E	k	$\frac{k}{\alpha^2 m_e c^2}$
Hartree/Bohr	Ha/Bohr	F	hartree/bohr	1
eV/Å*	eV/A	F	ev/ang	$\frac{e\hbar}{\alpha^3 m_e^2 c^3} \times 10^{10}$
Newton	N	F	n	$\frac{\hbar}{\alpha^3 m_e^2 c^3}$
dyne	dyne	F	dyne	$\frac{\hbar}{\alpha^3 m_e^2 c^3} \times 10^{-5}$

Table 4: The units of length, mass, time, charge energy and force in which physical data may be input and output. The dimensions for which unit conversion is provided are as follows: L = length, M = mass, T = time, C = charge, E = energy, F = force, V = velocity, P = pressure, 1/L = reciprocal length, F/L = Force constant and Vol = Volume. The default for each dimension is indicated by a *. The relative values are given in terms of SI fundamental constants.

Unit	Abbreviation	Dimension	Identifier	Value (atomic units)
Atomic Unit of Velocity	auv	V	auv	1
Ångstom/ps*	A/ps	V	ang/ps	$\frac{1}{c\alpha} \times 10^2$
Ångstom/fs	A/fs	V	ang/fs	$\frac{1}{c\alpha} \times 10^5$
Bohr/ps	Bohr/ps	V	bohr/ps	$\frac{\hbar}{c^2\alpha^2 m_e}^{12}$
Bohr/fs	Borh/fs	V	bohr/fs	$\frac{\hbar}{c^2\alpha^2 m_e}^{15}$
Metre/Second	m/s	V	m/s	$\frac{1}{c\alpha}$
Hartree/Bohr^3	Ha/Bohr**3	P	hartree/bohr**3	1
eV/Ångstrom^3	eV/A**3	P	ev/ang**3	$\frac{e\hbar^3}{\alpha^5 c^5 m_e^4} \times 10^{30}$
Pascal	Pa	P	pa	$\frac{\hbar^3}{\alpha^5 m_e^4 c^5}$
Megapascal	MPa	P	mpa	$\frac{\hbar^3}{\alpha^5 m_e^4 c^5} \times 10^6$
Gigapascal*	GPa	P	gpa	$\frac{\hbar^3}{\alpha^5 m_e^4 c^5} \times 10^9$
Atmosphere	Atm	P	atm	$\frac{101325.027\hbar^3}{\alpha^5 m_e^4 c^5} \times 10^6$
bar	bar	P	bar	$\frac{\hbar^3}{\alpha^5 m_e^4 c^5} \times 10^5$
Megabar	Mbar	P	mbar	$\frac{\hbar^3}{\alpha^5 m_e^4 c^5} \times 10^{11}$
Bohr$^-$1	1/Bohr	1/L	1/bohr	1
Metre$^-$1	1/m	1/L	1/m	$\frac{\hbar}{m_e c\alpha}$
Nanometre$^-$1	1/nm	1/L	1/nm	$\frac{\hbar}{m_e c\alpha} \times 10^9$
Ångstrom$^-$1*	1/A	1/L	1/ang	$\frac{\hbar}{m_e c\alpha} \times 10^{10}$
Hartree/Bohr^2	Ha/Bohr**2	F/L	ha/bohr**2	1
ev/Ångstrom^2*	ev/A**2	F/L	ev/ang**2	$\frac{e\hbar}{\alpha^3 m_e^2 c^3} \times 10^{10}$
Newton/metre	N/m	F/L	n/m	$\frac{\hbar^2}{\alpha^4 m_e^3 c^3}$
dyne/centimetre	dyne/cm	F/L	dyne/cm	$\frac{\hbar^2}{\alpha^4 m_e^3 c^3} \times 10^{-3}$
Bohr^3	Bohr**3	Vol	bohr**3	1
Metre^3	m**3	Vol	m**3	$\left(\frac{m_e c\alpha}{\hbar}\right)^3$
Centimetre^3	cm**3	Vol	cm**3	$\left(\frac{m_e c\alpha}{\hbar}\right)^3 \times 10^{-6}$
Nanometre^3	nm**3	Vol	nm**3	$\left(\frac{m_e c\alpha}{\hbar}\right)^3 \times 10^{-27}$
Angstrom^3	A**3	Vol	ang**3	$\left(\frac{m_e c\alpha}{\hbar}\right)^3 \times 10^{-30}$

Table 5: The units of velocity, pressure, reciprocal length, force constant and volume in which physical data may be input and output. The dimensions for which unit conversion is provided are as follows: L = length, M = mass, T = time, C = charge, E = energy, F = force, V = velocity, P = pressure, 1/L = reciprocal length, F/L = Force constant and Vol = Volume. The default for each dimension is indicated by a *. The relative values are given in terms of SI fundamental constants.

Keyword	Type	Corresponding Variable	Description
General Parameters			
COMMENT	S	`comment`	A comment string that may be used to lable the output.
IPRINT	I	`iprint`	Controls the verbosity of output.
STOP	D	–	If present, execution will halt the next time the parameter file is read.
CONTINUATION	S	`continuation`	The model file from which to continue a calculation.
REUSE	S	`reuse`	The model file from which to read data to initialise a new run.
CHECKPOINT	S	`checkpoint`	The model file to which model data should be written.
TASK	S	`task`	The job to be performed.
CALCULATE_STRESS	L	`calculate_stress`	If `TRUE` this forces the claculation of the stress tensor for any task.
RUN_TIME	I	`run_time`	The maximum run time for the job. If this is ≤ 0 no limit will be imposed.
NUM_BACKUP_ITER*	I	`num_backup_iter`	The number of geometry optimisation, MD or phonon iterations between writing backup restart files.
BACKUP_INTERVAL*	I	`backup_interval`	The interval in seconds between backups in geometry optimisation, MD or phonon.
PRINT_CLOCK	L	`print_clock`	Flag to indicate if timing information should be printed during the run.
LENGTH_UNIT	S	`length_unit`	The unit of length for output.
MASS_UNIT	S	`mass_unit`	The unit of mass for output.
TIME_UNIT	S	`time_unit`	The unit of time for output.
CHARGE_UNIT	S	`charge_unit`	The unit of charge for output.
ENERGY_UNIT	S	`energy_unit`	The unit of energy for output.
FORCE_UNIT	S	`force_unit`	The unit of force for output.
VELOCITY_UNIT	S	`velocity_unit`	The unit of velocity for output.
PRESSURE_UNIT	S	`pressure_unit`	The unit of pressure for output.
INV_LENGTH_UNIT	S	`inv_length_unit`	The unit of inverse length for output.
FREQUENCY_UNIT	S	`frequency_unit`	The unit of frequency for output.
FORCE_CONSTANT_UNIT	S	`force_constant_unit`	The unit of force constant for output.
VOLUME_UNIT	S	`volume_unit`	The unit of volume for output.

Table 6: Parameter file keywords controlling general parameters. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, and S for a text string.

* The keywords NUM_BACKUP_ITER and BACKUP_INTERVAL may not be defined in the same parameter file.

Keyword	Type	Corresponding Variable	Description
General Parameters Continued			
PAGE_WVFNS	I	`page_wvfns`	Controls the paging of large wavefunctions to disc in order to save memory.
RAND_SEED	I	`rand_seed`	Controls the initialisation of the random number sequence.
DATA_DISTRIBUTION	S	`data_distribution`	Controls the parallelisation strategy used by the code.
OPT_STRATEGY*	S	`opt_strategy`	Controls the optimisation strategy used by the code.
OPT_STRATEGY_BIAS*	I	`opt_strategy_bias`	Expert control for the optimisation strategy used by the code.
DEVEL_CODE	S	`devel_code`	A code for developers use to control specific debugging output.
Exchange-Correlation Parameters			
XC_FUNCTIONAL	S	`xc_functional`	The functional to use for the exchange-correlation potential. See Section 2.32 for details.
XC_VXC_DERIV_EPSILON	R	`xc_vxc_deriv_epsilon`	The fraction used to determine the size of the increment used in the numerical calculation of the second derivatives of the GGA functions.
Pseudopotentials			
PSPOT_NONLOCAL_TYPE	S	`pspot_nonlocal_type`	This defines the representation (real or reciprocal space) used for application of the non-local pseudopotential projectors.
PSPOT_BETA_PHI_TYPE	S	`pspot_beta_phi_type`	This defines the representation (real or reciprocal space) used for calculating the $< \beta \vert \phi >$ overlaps.

Table 7: Parameter file keywords controlling exchange-correlation, pseudopotential and basis set. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, and S for a text string.

* The keywords OPT_STRATEGY and OPT_STRATEGY_BIAS may not both be defined in the same parameter file.

Keyword	Type	Corresponding Variable	Description
Basis Set Parameters			
BASIS_PRECISION*	S	`basis_precision`	The accuracy of the basis set. See Section 2.36 for options.
CUT_OFF_ENERGY*	P	`cut_off_energy`	The cut off energy for the plane-wave basis set.
GRID_SCALE	R	`grid_scale`	The grid size as a multiple of the diameter of the plane-wave sphere.
FINE_GMAX	P	`fine_gmax`	Defines the size of the fine grid.
FINITE_BASIS_CORR	S	`finite_basis_corr`	Perform finite basis set correction when cell parameters change? See Section 2.39 for options.
BASIS_DE_DLOGE	P	`basis_de_dloge`	The derivative of total energy w.r.t. log of basis cut off energy. Only used if FINITE_BASIS_CORR = MANUAL
FINITE_BASIS_NPOINTS	I	`finite_basis_npoints`	The number of points used to estimate the derivative of total energy w.r.t. log of basis cut off energy. Only used if FINITE_BASIS_CORR = AUTO
FINITE_BASIS_SPACING	P	`finite_basis_spacing`	The energy difference between cut off energies at which total energy is evaluated to estimate the derivative of total energy w.r.t. log of basis cut off energy. Only used if FINITE_BASIS_CORR = AUTO
FIXED_NPW	L	`fixed_npw`	Flag to indicate if the basis set should be fixed during variable cell calculations.

Table 8: Parameter file keywords controlling the basis set. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, and S for a text string.

* BASIS_PRECISION and CUT_OFF_ENERGY may not both be defined in the same parameter file.

Keyword	Type	Corresponding Variable	Description
Electronic Parameters			
NELECTRONS*	I	nelectrons	The number of electrons in the system.
CHARGE*	I	–	The total system charge.
SPIN*	I	–	The total z-component of the electronic spin.
NUP*	I	nup	The number of spin up electrons.
NDOWN*	I	ndown	The number of spin down electrons.
SPIN_POLARISED*	L	–	Should the system be treated as spin polarised?
NBANDS†	I	nbands	The number of bands.
NEXTRA_BANDS†	I	–	The number of extra bands above $\lceil \frac{\texttt{nelectrons}}{2} \rceil$.
PERC_EXTRA_BANDS†	R	–	The number of extra bands as a percentage above $\lceil \frac{\texttt{nelectrons}}{2} \rceil$.
ELEC_TEMP	P	elec_temp	The electron temperature for which results will be calculated. Only used if metals_method='EDFT'.
EXCITED_STATE_SCISSORS	P	excited_state_scissors	Apply a correction to the band gap.
Electronic Minimisation Parameters			
ELECTRONIC_MINIMISER	S	–	The method of electronic minimisation to use. Possible values: "SD", "CG", "RMM/DIIS".
MAX_SD_STEPS	I	max_sd_steps	The maximum number of steepest descent steps in an SCF cycle.
MAX_CG_STEPS	I	max_cg_steps	The maximum number of conjugate gradient steps in an SCF cycle.
MAX_DIIS_STEPS	I	max_diis_steps	The maximum number of RMM/DIIS steps in an SCF cycle.
METALS_METHOD	S	metals_method	The method to be used for the treatment of metals or partial occupancies. See Section 2.54 for possible values.
ELEC_ENERGY_TOL	P	elec_energy_tol	The convergence tolerance for finding the ground state energy as an energy per atom.
ELEC_EIGENVALUE_TOL	P	elec_eigenvalue_tol	The convergence criteria for an eigenvalue during a band by band minimisation.

Table 9: Parameter file keywords controlling electronic and electronic minimisation parameters. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value, D for a keyword that may simply be defined (present) or not, and S for a text string. Groups of parameters which are mutually exclusive are indicated by integers in the Group column.

* Pairs of parameters defining the number of electrons and the total spin, eg. NUP and NDOWN or CHARGE and SPIN, may be defined together, but conflicting values must not be defined, eg. NELECTRONS and CHARGE.

† Only one of NBANDS, NEXTRA_BANDS and PERC_EXTRA_BANDS may be present in a parameter file.

Keyword	Type	Corresponding Variable	Description
Electronic Minimisation Parameters Continued			
ELEC_CONVERGENCE_WIN	I	elec_convergence_win	The number of over which convergence of the minimiser is assessed.
MAX_SCF_CYCLES	I	max_scf_cycles	The maximum number of SCF cycles in an electronic minimisation.
SPIN_FIX	I	spin_fix	The number of iterations to fix the spin during an electronic relaxation. If spin_fix $<$ 0, the spin is fixed permanently. Only used if fix_occupancy is FALSE.
FIX_OCCUPANCY	L	fix_occupancy	Fix the occupancies of the bands, ie. treat as a zero temperature insulator.
SMEARING_SCHEME	S	smearing_scheme	The smearing scheme to use for the fermi-surface of a metal. See Section 2.61 for details.
SMEARING_WIDTH	P	smearing_width	The width of the smearing of the fermi-surface of a metal.
EFERMI_TOL	P	efermi_tol	The tolerance within which to find the fermi energy of a metal.
NUM_OCC_CYCLES	I	num_occ_cycles	The number of occupancy minimisation cycles per electronic step during ensemble DFT minimisation.
ELEC_DUMP_FILE	S	elec_dump_file	The filename of the file into which to periodically dump the wavefunction and density during electronic minimisation as a backup.
NUM_DUMP_CYCLES	I	num_dump_cycles	The number of SCF cycles between wavefunction dumps.
ELEC_RESTORE_FILE	S	elec_restore_file	Restore the wavefunction and density from this file on first call to electronic minimisation.

Table 10: Parameter file keywords controlling electronic minimisation continued. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.

Keyword	Type	Corresponding Variable	Description
Density Mixing Parameters			
MIXING_SCHEME	S	`mixing_scheme`	The mixing scheme to use in the density mixing procedure.
MIX_HISTORY_LENGTH	I	`mix_history_length`	The number of charge densities to store in the density mixing history.
MIX_CHARGE_AMP	R	`mix_charge_amp`	The mixing amplitude for the charge density
MIX_CHARGE_GMAX	P	`mix_charge_gmax`	The maximum wavevector for which to mix the charge density.
MIX_SPIN_AMP	R	`mix_spin_amp`	The mixing amplitude for the spin density
MIX_SPIN_GMAX	P	`mix_spin_gmax`	The maximum wavevector for which to mix the spin density.
MIX_CUT_OFF_ENERGY	P	`mix_cut_off_energy`	The cut off energy for the densities within the mixing scheme.
MIX_METRIC_Q	P	`mix_metric_q`	The metric for the densities within the density mixing scheme.
Population Analysis			
POPN_CALCULATE	L	`popn_calculate`	Perform a population analysis on the final ground state?
POPN_BOND_CUTOFF	P	`popn_bond_cutoff`	The maximum distance between two atoms for which a bond population will be output.
PDOS_CALCULATE_WEIGHTS	L	`pdos_calculate_weights`	Calculate the band weights for a partial density of states analysis?

Table 11: Parameter file keywords controlling density mixing, and population analysis. Argument types are represented by, I for a integer, R for a real number, L for a logical value and S for a text string.

Keyword	Type	Corresponding Variable	Description
Geometry Optimisation Parameters			
GEOM_METHOD	S	`geom_method`	The method to use for geometry optimisation
GEOM_MAX_ITER	I	`geom_max_iter`	The maximum number of geometry optimisation iterations perform.
GEOM_ENERGY_TOL	P	`geom_energy_tol`	The convergence tolerance for the total energy per atom when finding ground state structure.
GEOM_FORCE_TOL	P	`geom_force_tol`	The convergence tolerance for the maximum force on the ions when finding the ground state ionic positions.
GEOM_DISP_TOL	P	`geom_disp_tol`	The convergence tolerance for the maximum ionic displacement in a step when finding the ground state ionic positions.
GEOM_STRESS_TOL	P	`geom_stress_tol`	The convergence tolerance for the maximum stress when finding the ground state cell parameters
GEOM_CONVERGENCE_WIN	I	`geom_convergence_win`	The number of geometry optimisation iterations during which the convergence criteria must be met for convergence to be accepted.
GEOM_MODULUS_EST	P	`geom_modulus_est`	An estimate of the bulk modulus of the system.
GEOM_FREQUENCY_EST	P	`geom_frequency_test`	An estimate of the average phonon frequency at the gamma point.

Table 12: Parameter file keywords controlling geometry optimisation. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.

Keyword	Type	Corresponding Variable	Description
Band Structure Parameters			
BS_MAX_ITER	I	`bs_max_iter`	The maximum number of iterations when calculating the band structure.
BS_EIGENVALUE_TOL	P	`bs_eigenvalue_tol`	The convergence criteria for an eigenvalue during a band structure calculation.
BS_MAX_CG_STEPS	I	`bs_max_cg_steps`	The maximum number of conjugate gradient steps in the band structure minimisation before resetting.
BS_NEXTRA_BANDS*	I	–	The number of extra bands above the number of valence bands at each k-point when calculating the band structure.
BS_PERC_EXTRA_BANDS*	R	–	The precentage of extra bands above the number of valence bands at each k-point when calculating the band structure.
BS_NBANDS*	I	`bs_nbands`	The number of bands at each k-point when calculating the band structure.
BS_XC_FUNCTIONAL	S	`bs_xc_functional`	The exchange-correlation functional to use for band-structure calculations.
Molecular Dynamics			
MD_NUM_ITER	I	`md_num_iter`	The number of MD time steps.
MD_DELTA_T	P	`md_delta_t`	The time step for molecular dynamics
MD_ENSEMBLE	S	`md_ensemble`	The ensemble for the molecular dynamics run.
MD_TEMPERATURE	P	`md_temperature`	The temperature for the molecular dynamics run.
MD_THERMOSTAT	S	`md_thermostat`	The thermostat for the molecular dynamics run.
MD_NOSE_T	T	`md_nose_t`	The value for the characteristic Nosé-Hoover time. Only used if Nosé-Hoover thermostat has been chosen.
MD_LANGEVIN_T	P	`md_langevin_t`	The damping time for the Langevin thermostat. Only used if the Langevin thermostat has been chosen.

Table 13: Parameter file keywords controlling band structure and molecular dynamics. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.
* Only one of BS_NBANDS, BS_NEXTRA_BANDS and BS_PERC_EXTRA_BANDS may be present in a parameter file.

Keyword	Type	Corresponding Variable	Description
Molecular Dynamics Continued			
MD_EXTRAP	S	md_extrap	The extrapolation scheme to use for molecular dynamics.
MD_EXTRAP_FIT	L	md_extrap_fit	Use best fit extrapolation parameters?
MD_DAMPING_SCHEME	S	md_damping_scheme	Controls the scheme used for damped MD geometry optimisation.
MD_DAMPING_RESET	I	md_damping_reset	Reset the damping factors after this number of iterations if convergence has not been achieved.
MD_OPT_DAMPED_DELTA_T	L	md_opt_damped_delta_t	Use optimised time step for damped molecular dynamics?
MD_ELEC_ENERGY_TOL	P	md_elec_energy_tol	The convergence tolerance for finding the ground state energy during an MD run as an energy per atom.
MD_ELEC_EIGENVALUE_TOL	P	md_elec_eigenvalue_tol	The convergence criteria for an eigenvalue when performing DIIS/DM minimisation during an MD run.
MD_ELEC_CONVERGENCE_WIN	I	md_elec_convergence_win	The number of iterations over which convergence is assessed during an electronic minimsation in an MD run.

Table 14: Parameter file keywords controlling molecular dynamics continued. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.

Keyword	Type	Corresponding Variable	Description
Optics			
OPTICS_NEXTRA_BANDS*	I	–	The number of extra bands above the number of valence bands at each k-point when calculating an optical spectrum.
OPTICS_PERC_EXTRA_BANDS*	R	–	The precentage of extra bands above the number of valence bands at each k-point when calculating an optical spectrum.
OPTICS_NBANDS*	I	`optics_nbands`	The number of bands at each k-point when calculating an optical spectrum.
OPTICS_XC_FUNCTIONAL	S	`optics_xc_functional`	The functional to use for calculating optical spectra.
Transition State Search			
TSSEARCH_METHOD	S	`tssearch_method`	Determines the method used in the transitionstate search algorithm
TSSEARCH_LSTQST_PROTOCOL	S	`tssearch_lstqst_protocol`	Determines the protocol used when performing and LST/QST search.
TSSEARCH_QST_MAX_ITER	I	`tssearch_qst_max_iter`	The maximum number of iterations in a QST search.
TSSEARCH_CG_MAX_ITER	I	`tssearch_cg_max_iter`	The maximum number of conjugate gradients iterations in the transistion state search.
TSSEARCH_FORCE_TOL	P	`tssearch_force_tol`	The force tolerance within which the transition state will be found.
TSSEARCH_DISP_TOL	P	`tssearch_disp_tol`	The displacement tolerance within which the transition state will be found.

Table 15: Parameter file keywords controlling optics and transition state searches. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.
* Only one of OPTICS_NBANDS, OPTICS_NEXTRA_BANDS and OPTICS_PERC_EXTRA_BANDS may be present in a parameter file.

Keyword	Type	Corresponding Variable	Description
Phonon			
PHONON_MAX_CYCLES	I	phonon_max_cycles	The maximum number of cycles in the minimisation algorithm.
PHONON_MAX_CG_STEPS	I	phonon_max_cg_steps	The maximum number of conjugate-gradient steps in a cycle of the phonon LR minimiser.
PHONON_ENERGY_TOL	P	phonon_energy_tol	The convergence tolerance for the phonon LR minimiser in units of force constant.
PHONON_CONVERGENCE_WIN	I	phonon_convergence_win	The number of over which convergence of the phonon LR minimiser is assessed.
PHONON_PRECONDITIONER	S	phonon_preconditioner	The preconditioner to use in the minimisation algorithm.
PHONON_USE_KPOINT_SYMMETRY	L	phonon_use_kpoint_symmetry	Use the irreducible k-point set of the (reduced) symmetry for phonon calculations?
PHONON_METHOD	S	phonon_method	The method to use to calculate the dynamical matrix.
PHONON_DOS_SPACING	P	phonon_dos_spacing	The resolution of the phonon density of states calculation.
PHONON_FINITE_DISP	P	phonon_finite_disp	The amplitude of the perturbation used in a finite displacement phonon calculation.
PHONON_CALC_LO_TO_SPLITTING	L	phonon_calc_lo_to_splitting	Whether to calculate the NS term of the dynamical matrix.
PHONON_SUM_RULE	L	phonon_sum_rule	Enforce phonon sum rule at q=0?
CALCULATE_BORN_CHARGES	L	calculate_born_charges	Calculate Born effective charges? (N.B. Also affects Efield calculations)
BORN_CHARGE_SUM_RULE	L	born_charge_sum_rule	Enforce Born charge sum rule?

Table 16: Parameter file keywords controlling phonon calculations. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.

Keyword	Type	Corresponding Variable	Description
Electric Field Linear Response			
EFIELD_MAX_CYCLES	I	`efield_max_cycles`	The maximum number of cycles in the minimisation algorithm.
EFIELD_MAX_CG_STEPS	I	`efield_max_cg_steps`	The maximum number of conjugate-gradient steps in a cycle of the efield LR minimiser.
EFIELD_ENERGY_TOL	P	`efield_energy_tol`	The convergence tolerance for the efield LR minimiser in units of force constant.
EFIELD_CONVERGENCE_WIN	I	`efield_convergence_win`	The number of over which convergence of the efield LR minimiser is assessed.
EFIELD_CALC_ION_ PERMITTIVITY	L	`efield_calc_ion_ permittivity`	Calculate the ionic permittivity when performing an efield calculation?
EFIELD_IGNORE_MOLEC_MODES	S	`efield_ignore_molec_modes`	Indicates if lowest modes should be ignored for molecules.
Deprecated			
PHONON_CONST_BASIS	L	–	No-longer used.

Table 17: Parameter file keywords controlling electric field linear response calculations and deprectaed keywords. Argument types are represented by, I for a integer, R for a real number, P for a physical value, L for a logical value and S for a text string.

2.7 character(len=file_maxpath) :: continuation

The model file from which the job will be continued. If not a continuation, this parameter should be 'NULL'.

If the keyword CONTINUATION is set to 'DEFAULT' (not case sensitive), this parameter will be set to '`seedname`.check'.

The default value of this parameter is 'NULL'.

2.8 character(len=file_maxpath) :: reuse

The model file from which data will be read to initialise a new calculation. If no file is to be used this parameter should be 'NULL'.

If the keyword REUSE is set to 'DEFAULT' (not case sensitive), this parameter will be set to '`seedname`.check'.

The default value of this parameter is 'NULL'.

2.9 character(len=file_maxpath) :: checkpoint

The file to which checkpoint data will be written.

The default value of this parameter is '`seedname`.check'.

2.10 character :: task

This defines the calculation performed. There are seven valid values of this parameter:

- SinglePoint
- BandStructure
- GeometryOptimization
- MolecularDynamics
- Phonon
- Efield
- Phonon+Efield
- Optics
- TransitionStateSearch

The default value of this parameter is 'SinglePoint'.

2.11 logical :: calculate_stress

If `TRUE` this forces the calculation of the stress tensor for any task. Otherwise, the stress will only be clalculated if required, e.g. during a cell geometry optimisation with cell relaxation.

The default value of this parameter is `FALSE`.

2.12 integer :: run_time

This specifies the maximum run time for the job. If `run_time` is greater than zero, the job will, if possible, exit cleanly before this time in seconds has elapsed, leaving as little unused time as possible.

Clean termination before `run_time` cannot be guaranteed, and the shortest operation which can be timed is an electronic minimisation or a single molecular dynamics, geometry optimisation or phonon step.

If `run_time` ≤ 0 no time limit will be imposed on the run.

The default value of this parameter is 0.

2.13 integer :: num_backup_iter

The number of geometry optimisation or MD iterations or Phonon q-points between writing backup restart files.

The default value of this parameter is 5, unless the user has explicitly set `backup_interval` to 0 in the .param file (indicating an infinite interval between backups), in which case no backups will be performed.

2.14 integer :: backup_interval

If a potential backup pooint is reached and more than `backup_interval` seconds have elapsed since the last backup, a backup will be forced. If `backup_interval` is 0, no timed backups will be performed. However backups will still take place every `num_backup_iter` iterations.

The default value of this parameter is 0.

2.15 logical :: print_clock

Flag to indicate if timing information should be printed during the run.

The default value of this parameter is `TRUE`.

2.16 character :: length_unit

The units in which lengths will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'ang'.

2.17 character :: mass_unit

The units in which masses will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'amu'.

2.18 character :: time_unit

The units in which times will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'ps'.

2.19 character :: charge_unit

The units in which charges will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'e'.

2.20 character :: energy_unit

The units in which energies will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'ev'.

2.21 character :: force_unit

The units in which forces will be output. The possible values of this parameter are listed in Table 4.

The default value of this parameter is 'ev/ang'.

2.22 character :: velocity_unit

The units in which pressures will be output. The possible values of this parameter are listed in Table 5.

The default value of this parameter is 'ang/ps'.

2.23 character :: pressure_unit

The units in which pressures will be output. The possible values of this parameter are listed in Table 5.

The default value of this parameter is 'gpa'.

2.24 character :: inv_length_unit

The units in which inverse lengths will be output. The possible values of this parameter are listed in Table 5.

The default value of this parameter is '1/ang'.

2.25 character :: frequency_unit

The units in which frequencies will be output. The possible values of this parameter are listed in table 4 and are the same as those in which energies may be output.

The default value of is parameter is 'CM-1'.

2.26 character(len=30) :: force_constant_unit

The units in which force constants will be output. The possible values of this parameter are listed in table 5.

The default value of this parameter is 'ev/ang**2'.

2.27 character(len=30) :: volume_unit

The units in which volumes will be output. The possible values of this parameters are listed in table 5.

The default value of this parameter is 'A**3'.

2.28 integer :: page_wvfns

This controls the paging of wavefunctions to disc in order to save memory. If this variable is 0, no paging will be performed. If less than 0, all wavefunctions will be paged to disc. If the value of this variable is greater than 0, all wavefunctions requiring more memory than this value in megabytes will be paged to disc.

The default value of this parameter is 0.

2.29 integer :: rand_seed

This controls the initialisation of the random number sequence. If `rand_seed` is 0, the seed for the random number generation is selected pseudorandomly. If this parameter is non-zero, the value is used as a seed for the random number generator.

The default value of this parameter is 0.

2.30 character :: data_distribution

This parameter determines the parallelisation strategy used. The possible values are:

- Kpoint - only k-point parallelisation will be used.
- Gvector - only g-vector parallelisation will be used.
- Mixed - a combination of k-point and g-vector parallelisation will be used.
- Default - The optimal parallelisation strategy for the architecture will be used.

The default value of this parameter is 'Default'.

2.31 character :: opt_strategy

This parameter determines the optimisation strategy used, where multiple strategies may be used for an algorithm, with differing costs in terms of memory usage or performance.

Possible values are:

- Speed - Use the optimisation strategy which maximises performance at the cost of additional memory usage.
- Default - Use the optimisation strategy that balances performance and memory usage.
- Memory - Use the optimisation strategy which minimises memory usage at a cost of decreased performance.

2.32 character :: xc_functional

The functional to use when calculating the exchange-correlation potential. Possible values are:

- LDA - Local Density Approximation
- PW91 - Perdew Wang '91 GGA
- PBE - Perdew Burke Ernzerhof
- RPBE - Revised Perdew Burke Ernzerhof
- HF - Hartree-Fock
- SHF - Screened Hartree-Fock
- EXX - Exact exchange
- SX - Screened exchange
- ZERO - No exchange-correlation potential
- HF-LDA -LDA with exchange contribution replaced by Hartree-Fock
- SHF-LDA - -LDA with exchange contribution replaced by screened Hartree-Fock
- EXX-LDA - -LDA with exchange contribution replaced by exact exchange
- SX-LDA - LDA with exchange contribution replaced by screeened exchange

The default value of this parameter is 'LDA'.

2.33 character :: pspot_nonlocal_type

This controls the representation of the non-local part of the pseudopotential. Possible values are:

- RECIPROCAL - reciprocal space non-local pseudopotentials
- REAL - real space non-local pseudopotentials

The default value of this parameter is 'RECIPROCAL'.

2.34 character :: pspot_beta_phi_type

This controls the representation of the non-local part of the pseudopotential used for calculation of the $< \beta|\phi >$ overlaps. Possible values are:

- RECIPROCAL - reciprocal space non-local pseudopotentials
- REAL - real space non-local pseudopotentials

This parameter can only take the value 'REAL' if `pspot_nonlocal_type` is 'REAL'.
The default value of this parameter is `pspot_nonlocal_type`.

2.35 `character :: basis_precision`

This specifies the precision of the basis set by choosing the level of convergence of atomic energies with respect to the plane wave cut off energy for the pseudopotentials used in the calculation.

Options for BASIS_PRECISION are

- COARSE - 1 eV per atom
- MEDIUM - 0.3 eV per atom
- FINE - 0.1 eV per atom
- PRECISE - 1.2×FINE
- EXTREME - 0.01 eV per atom.

The default value for this parameter is 'FINE'.

If CUT_OFF_ENERGY is specified in the parameter file, this parameter will be set to 'NULL'.

2.36 `real :: cut_off_energy`

This holds the cut off energy for the plane wave basis set being used in the current calculation.

If the BASIS_PRECISION keyword is defined in the parameter file, the cut off energy will be equal to the highest of the cut off energies associated with the chosen level of accuracy for the pseudopotentials used in the calculation. This will be set by the `ion_initialise` subroutine.

If neither BASIS_PRECISION nor CUT_OFF_ENERGY are defined in the parameter file, the default cut off energy is that associated with the FINE level of accuracy for the pseudopotentials in the calculation.

If the CUT_OFF_ENERGY keyword is not present in the input file, this parameter will be set to -1 Ha until the correct value is assigned by the `ion_initialise` subroutine.

2.37 `real :: grid_scale`

The size of the standard grid as a multiple of the diameter of cut off sphere.

The default value will be 1.75.

2.38 `real :: fine_gmax`

The fine grid will be chosen to be of the minimum size such that all g-vectors with $|\mathbf{g}| \leq$ `fine_gmax` are included in the fine grid.

If not specified, `fine_gmax` will be set to -1 ${a_0}^{-1}$, resulting in the fine and standard grids being identical. In this case, the correct value of `fine_gmax` will be set by the subroutine `basis_initialise`.

2.39 `integer :: finite_basis_corr`

This determined whether finite basis set corrections to the energy and stress will be performed. Possible values are

- 0 - No correction performed
- 1 - Correction performed with user provided estimate of $\frac{dE_{tot}}{d\log(E_{cut})}$.
- 2 - Correction performed with automatic calculation of $\frac{dE_{tot}}{d\log(E_{cut})}$.

The vlaue of this parameter is defined by the string parameter file keyword FINITE_BASIS_CORR as follows:

- NONE - 0
- MANUAL - 1

- AUTO - 2

For backwards compatability, the string "0" is equivalent to "NONE", "1" to "MANUAL" and "2" to "AUTO".

If FINITE_BASIS_CORR="MANUAL", the user must supply a value for keyword BASIS_DE_DLOGE.

The default value of this parameter is 2 if one of the following conditions are met:

- `task`=GEOMETRYOPTIMIZATION, `geom_method`=BFGS, `fixed_npw`=`FALSE` and the cell is not fixed; or
- `task`=MOLECULARDYNAMICS, `fixed_npw`=`FALSE` and the cell is not fixed; or
- `calculate_stress` = `TRUE`.

Otherwise, the default value is 0.

2.40 `real(kind=dp) :: basis_de_dloge`

The derivative of total energy with respect to the log of basis cut-off energy.

This is only used if `finite_basis_corr` = "MANUAL"

The default value of this parameter is 0.0 eV.

2.41 `integer :: finite_basis_npoints`

The number of points used to estimate the derivative of total energy with respect to the log of basis cut off energy. The minimum value allowed for this parameter is 2.

The cut-off energies at which the total energy is calculated, in order to estimate the derivative, will be chosen equally spaced by `finite_basis_spacing` with the highest energy equal to `cut_off_energy`.

The default value of this parameter is 3.

Only used if `finite_basis_corr` = "AUTO"

2.42 `real(kind=dp) :: finite_basis_spacing`

The energy difference between cut-off energies at which total energy is evaluated in order to estimate the derivative of total energy with respect to the log of basis cut-off energy.

The cut-off energies at which the total energy is calculated, in order to estimate the derivative, will be chosen symmetrically about the chosen cut-off energy for the calculation.

The default value of this parameter is 5 eV.

Only used if `finite_basis_corr` = "AUTO"

2.43 `logical :: fixed_npw`

Logical flag to indicate if the basis should be fixed during variable cell calculations. The alternative is to allow the number of basis states to vary while fixing the cut-off energy.

The default value of this parameter is `TRUE`.

2.44 `integer :: nelectrons`

The total number of electrons in the system.

If the CHARGE keyword is specified, `nelectrons` will be chosen such that the total system charge is equal to the argument of CHARGE.

Alternatively, if NUP and NDOWN are specified, the `nelectrons` will be the sum of the arguments of NUP and NDOWN.

If the number of electrons is not specified in the parameter file, the default value will be chosen such that the system is charge neutral.

2.45 integer :: nup

The number of spin up electrons.

If SPIN has been specified then nup $= \frac{\texttt{nelectrons}+\text{SPIN}}{2}$.

If neither NUP nor SPIN have been specified, nup $= \lceil \frac{\texttt{nelectrons}}{2} \rceil$.

2.46 integer :: ndown

The number of spin down electrons.

If SPIN has been specified then ndown $= \frac{\texttt{nelectrons}-\text{SPIN}}{2}$.

If neither NDOWN nor SPIN have been specified, ndown $= \lfloor \frac{\texttt{nelectrons}}{2} \rfloor$.

2.47 integer :: nspins

The number of spin components.

If the keyword SPIN_POLARIZED is present this determines the value of nspins. If SPIN_POLARIZED = FALSE then nspins = 1, otherwise nspins = 2.

Default, 1 if nup = ndown, otherwise 2.

2.48 integer :: nbands

The maximum number of bands at any k-point and spin.

If NBANDS is present in the parameter file, the value of nbands is determined by this keyword.

If NEXTRA_BANDS is specified nbands = max(nup, ndown) + NEXTRA_BANDS.

If PERC_EXTRA_BANDS is specified nbands = max(nup, ndown) $\times$ $(1 + \frac{\text{PERC_EXTRA_BANDS}}{100})$.

If NBANDS, NEXTRA_BANDS and PERC_EXTRA_BANDS have not been specified and fix_occupancy=TRUE then nbands = max(nup, ndown) .If fix_occupancy=FALSE or elec_temp > 0, the default value of nbands will be the ceiling of the default value multiplied by 1.2.

2.49 real :: elec_temp

The electron temperature for which to calculated results. This will only be used if metals_method='EDFT'.

The default value of this parameter is 0K.

2.50 real(kind=dp) :: excited_state_scissors

Apply the "scissors" operator to add an offset to conduction-band eigenvalues as empirical correction for LDA/GGA underestimation of band-gaps.

The default value of this paramter is 0.0 eV.

2.51 integer :: max_sd_steps

The maximum number of steepest descent steps in an SCF cycle.

If not explicitly set in the parameter file, the value of this parameter depends on the value of the keyword ELECTRONIC_MINIMISER. The values of this parameter for each value of ELECTRONIC_MINIMISER are:

- SD : 10
- CG : 1
- RMM/DIIS : 1

If keyword ELECTRONIC_MINIMISER is not present in the parameter file, the default value will be 1.

2.52 integer :: max_cg_steps

The maximum number of conjugate gradient steps in an SCF cycle.

If not explicitly set in the parameter file, the value of this parameter depends on the value of the keyword ELECTRONIC_MINIMISER. The values of this parameter for each value of ELECTRONIC_MINIMISER are:

- SD : 0
- CG : 4
- RMM/DIIS : 2

If keyword ELECTRONIC_MINIMISER is not present in the parameter file, the default value will be 4.

2.53 integer :: max_diis_steps

The maximum number of RMM/DIIS steps in an SCF cycle.

If not explicitly set in the parameter file, the value of this parameter depends on the value of the keyword ELECTRONIC_MINIMISER. The values of this parameter for each value of ELECTRONIC_MINIMISER are:

- SD : 0
- CG : 0
- RMM/DIIS : 7

If keyword ELECTRONIC_MINIMISER is not present in the parameter file, the default value will be 0.

2.54 character :: metals_method

This determines the method used in calculations on metals.

The possible values are:

- NONE - no special treatment of metals.
- DM - metals treated by density mixing.
- EDFT - metals treated by Ensemble DFT.

These methods may be useful in the treatment of non-metalic systems in some cases.

If `fix_occupancy=FALSE` the default value of this parameter is EDFT, otherwise it is NONE.

2.55 real :: elec_energy_tol

The tolerance for accepting convergence of the total energy in an electronic minimisation.

The default value for this parameter is 1×10^{-5}eV per atom.

2.56 real :: elec_eigenvalue_tol

The tolerance for accepting convergence of a single eigenvalue during band-by-band minimisation.

The default value for this parameter is $\texttt{min}(\frac{\texttt{elec_energy_tol} \times \texttt{num_atoms}}{\texttt{nbands}}, 1 \times 10^{-6})$ eV. Here `num_atoms` is passed as an argument to `parameters_read`.

2.57 integer :: elec_convergence_win

The total energy or eigenvalue must lie within `convergence_tol` or `elec_eigenvalue_tol` respectively for the last `elec_energy_win` iterations for the convergence criteria to be met.

The value of `elec_convergence_win` must be greater than or equal to 2.

The default value of this parameter is 3.

2.58 integer :: max_SCF_cycles

The maximum number of SCF cycles performed in an electronic minimisation. The electronic minimisation will end regardless of whether the convergence criteria have been met after max_SCF_cycles SCF cycles.

The default value of this parameter is 30.

2.59 integer :: spin_fix

The number of electronic iterations for which the total spin is fixed. If spin_fix < 0, the spin will be fixed for the duration of the calculation. Only used if fix_occupancy = FALSE, and spin_polarized = TRUE. For insulators the spin is fixed regardless of the value of cspin_fix.

The default value of this parameter is 10.

2.60 logical :: fix_occupancy

A logical flag which indicates whether the occupancies of the bands should be fixed, i.e. if the system should be treated as a zero temperature insulator.

The default value of this parameter is FALSE.

2.61 character :: smearing_scheme

This indicates the smearing scheme to use if treating the system as a metal. This is only used if fix_occupancy = FALSE.

The valid options for this parameter are:

- Gaussian
- GaussianSplines
- FermiDirac
- HermitePolynomials
- ColdSmearing

The default value for smearing_scheme is 'Gaussian'.

2.62 real :: smearing_width

The width of smearing of the Fermi surface, if treating the system as a metal. This is only used if fix_occupancy = FALSE.

The default value of this parameter is 0.2 eV.

2.63 real :: efermi_tol

The tolerance within which the Fermi energy is calculated for a metallic system or finite temperature insulator. This is only used if fix_occupancy = FALSE or elec_temp > 0.

The default value of this parameter is $0.1 \times$ elec_eigenvalue_tol.

2.64 integer :: num_occ_cycles

The number of occupancy cycles performed for each wavefunction minimisation step during an Ensemble DFT run. This is only used if fix_occupancy = FALSE or elec_temp > 0 and metals_method = EDFT.

The default value of this parameter is 6.

2.65 chracter(len=file_maxpath) :: elec_dump_file

The filename of the file into which to periodically dump the wavefunction and density during electronic minimisation. This may be used as a backup and restored using the `elec_restore_file` parameter.

If this parameter is set to 'NULL', no backup wavefunction will be written.

The default wave of this parameter is '`seedname`.wvfn'.

2.66 integer :: num_dump_cycles

The number of SCF cycles between wavefunction dumps.

If `num_dump_cycles` ≤ 0, no wavefunction dumps will be performed.

The default value of this parameter is 5.

2.67 character(len=file_maxpath) :: elec_restore_file

The wavefunction and density will be restored from this file on first call to electronic minimisation. The basis set and distribution must be unchanged from the run in which the electronic file was written.

If this parameter is set to 'NULL' the wavefunction and density will not be read.

The default value of this parameter is 'NULL'.

2.68 character :: mixing_scheme

This determines the mixing scheme to be used in the density mixing procedure. The possible values are

- Kerker
- Linear
- Broyden
- Pulay

The default value of this parameter is 'BROYDEN'.

2.69 integer :: mix_history_length

The maximum number of densities to store in the history used in the density mixing procedure.

The default value of this parameter is 7.

2.70 real :: mix_charge_amp

The mixing amplitude for the charge density in the density mixing procedure.

The default value of this parameter is 0.8.

2.71 real :: mix_charge_gmax

The maximum g-vector at which the charge density is mixed in the density mixing procedure.

The default value of this parameter is 1.5 Å^{-1}.

2.72 real :: mix_spin_amp

The mixing amplitude for the spin density in the density mixing procedure.

The default value of this parameter is 2.0.

2.73 real :: mix_spin_gmax

The maximum g-vector at which the spin density is mixed in the density mixing procedure.

The default value of this parameter is 1.5 Å^{-1}.

2.74 real :: mix_cut_off_energy

This determines the extent of the grid used for mixing old and new densities. Density components with wave vectors corresponding to energies higher than this will not be mixed.

The default value is of this parameter is cut_off_energy.

2.75 real :: mix_metric_q

This determines the metric within the density mixing scheme. A weighting factor is used when evaluating scalar products of densities:

$$f(q) = \frac{(q^2 + \texttt{mix_metric_q}^2)}{q^2}$$

The default value of this parameter is -1, resulting in an appropriate value being set by the density mixing algorithm.

2.76 logical :: popn_calculate

This indicates if a population analysis should be performed on the final ground state of the calculation.

The default value of this paameter is TRUE.

2.77 real :: popn_bond_cutoff

This is the maximum distance between two atoms for which a bond population will be output when performing a population analysis.

The default value of this parameter is 3 Å.

2.78 logical :: pdos_calculate_weights

This indicates if the weight of each band in each localised orbital should be calculated on the final ground state of the calculation in order to allow a partial density of states analysis.

The default value of this parameter is FALSE.

2.79 integer :: bs_max_iter

The maximum number of iterations to perform when performing a band structure calculation.

The default value of this parameter is 60.

2.80 real :: bs_eigenvalue_tol

The tolerance for accepting convergence of a single eigenvalue during a band structure calculation.

The default value for this parameter is 1×10^{-6}eV.

2.81 integer :: bs_max_cg_steps

The maximum number of conjugate gradient steps to perform during a band structure calculations before resetting to the steepest descent direction.

The default value of this parameter is 4.

2.82 integer :: bs_nbands

The number of bands at each k-point when performing a band structure calculation.

If BS_NBANDS is present in the parameter file, the value of bs_nbands is determined by this keyword.

If BS_NEXTRA_BANDS is specified bs_nbands = max(nup, ndown) + BS_NEXTRA_BANDS.

If BS_PERC_EXTRA_BANDS is specified bs_nbands = max(nup, ndown)$\times(1+\frac{\text{BS_PERC_EXTRA_BANDS}}{100})$.

The default value of this parameter is $3 \times$ nbands.

2.83 character(len=15) :: bs_xc_functional

The exchange-correlation functional to use for band structure calculations. Possible values are:

- LDA - Local Density Approximation
- PW91 - Perdew Wang '91 GGA
- PBE - Perdew Burke Ernzerhof
- RPBE - Revised Perdew Burke Ernzerhof
- HF - Hartree-Fock
- SHF - Screened Hartree-Fock
- EXX - Exact exchange
- SX - Screened exchange
- ZERO - No exchange-correlation potential
- HF-LDA -LDA with exchange contribution replaced by Hartree-Fock
- SHF-LDA - -LDA with exchange contribution replaced by screened Hartree-Fock
- EXX-LDA - -LDA with exchange contribution replaced by exact exchange
- SX-LDA - LDA with exchange contribution replaced by screeened exchange

The default value of this parameter is the value of `xc_functional`.

2.84 character :: geom_method

The method of geometry optimisation to use. Possible values are

– BFGS - BFGS minimisation

– DampedMD - Damped molecular dynamics

The default value of this parameter is 'BFGS'.

2.85 integer :: geom_max_iter

The maximum number of geometry optimisation steps to take.
The default value of this parameter is 30.

2.86 real :: geom_energy_tol

The tolerance for finding convergence of the appropriate free energy per atom during a geometry optimisation. The difference between the maximum and minimum values of the free energy over `geom_convergence_win` iterations must be less than this value for the convergence criteria to have been met.
The default value of this parameter is 2×10^{-5} eV/atom.

2.87 real :: geom_force_tol

The tolerance for the ionic force to accept convergence during an ionic relaxation. The maximum ionic force must be less than this value over `geom_convergence_win` iterations for the convergence criteria to have been met.
The default value of this parameter is 0.05 eVÅ^{-1}.

2.88 real :: geom_disp_tol

The tolerance for the ionic displacement to accept convergence during an ionic relaxation. The maximum ionic displacement must be less than this value over `geom_convergence_win` iterations for the convergence criteria to have been met.

The default value of this parameter is 0.001 Å.

2.89 real :: geom_stress_tol

The tolerance for the stress to accept convergence during a cell relaxation. The maximum stress component must be less than this value over `geom_convergence_win` iterations for the convergence criteria to have been met.

The default value of this parameter is 0.1 GPa.

2.90 integer :: geom_convergence_win

The number of geometry optimisation iterations during which the convergence criteria must be met for convergence to be accepted.

The default value of this parameter is 2.

2.91 real :: geom_modulus_est

An estimate of the bulk modulus of the system. Used to initialise the Hessian for BFGS geometry optimisation with cell relaxation.

The default value of this parameter is 500 GPa.

2.92 real :: geom_frequency_est

An estimate of the average phonon frequency at the gamma point. Used to initialise the Hessian for BFGS geometry optimisation with ionic relaxation.

The default value of this parameter is 50 THz.

2.93 integer :: md_num_iter

The number of molecular dynamics iterations to perform.

The default value of this parameter is 100.

2.94 real :: md_delta_t

The time step for molecular dynamics.

The default value of this parameter is 1 fs.

2.95 character :: md_ensemble

The ensemble to use for molecular dynamics. The options are:

- NVT
- NVE

The default value for this parameter is 'NVE'.

2.96 real :: md_temperature

The temperature for molecular dynamics.

The default value for this parameter is 300K.

2.97 character :: md_thermostat

The thermostat to use for molecular dynamics. The options are:

- Nosé-Hoover
- Langevin

The default value for this parameter is 'Nosé-Hoover'.

2.98 real :: md_nose_t

The characteristic time for the Nosé parameter when using the the Nosé-Hoover thermostat. The Nosé mass is calculated from this.

This parameter is unused if the Nosé-Hoover thermostat has not been selected.

The default value of this parameter is $10 \times$ `md_delta_t`.

2.99 real :: md_langevin_t

The Langevin damping time when using the Langevin thermostat. This parameter is unused if the Langevin thermostat has not been selected.

The default value of this parameter is $100 \times$ `md_delta_t`.

2.100 character :: md_extrap

The extrapolation scheme to use for wavefunction, and charge density if using density mixing, during molecular dynamics, including damped MD. The options are:

- None - No extrapolation used.
- First - First order extrapolation.
- Second - Second order extrapolation.
- Mixed - Alternating first and second order extrapolation.

The default value of this parameter is 'First'.

2.101 logical :: md_extrap_fit

If `md_extrap_fit` = `TRUE`, the best extrapolation parameters will be calculated at each iteration. Otherwise, fixed extrapolation parameters will be used.

The default value of this parameter is `TRUE`.

2.102 character :: md_damping_scheme

This controls the damping scheme used during damped MD geometry optimisation. The possible values for this parameter are:

- Independent
- Coupled
- SteepestDescents

The default value of this parameter is 'Independent'.

2.103 integer :: md_damping_reset

Reset the damping factors after this number of iterations if convergence has not been achieved. If this parameter is set to 0, the damping factors will never be reset.

The default value of this parameter is 30.

2.104 logical :: md_opt_damped_delta_t

If md_opt_damped_delta_t = TRUE, the best optimal time step will be calculated at each iteration during damped molecular dynamics. Otherwise, a fixed time step will be used.

The default value of this parameter is FALSE.

2.105 real :: md_elec_energy_tol

The tolerance for accepting convergence of the total energy in an electronic minimisation during an MD run.

The default value for this parameter is the same as elec_energy_tol.

2.106 real :: md_elec_eigenvalue_tol

The tolerance for accepting convergence of a single eigenvalue during DIIS minimisation while performing an MD run.

The default value for this parameter is elec_eigenvalue_tol.

2.107 integer :: md_elec_convergence_win

The total energy or eigenvalue must lie within md_elec_energy_tol or md_elec_eigenvalue_tol respectively for the last md_elec_convergence_win iterations for the convergence criteria to be met during an MD run.

The value of md_elec_convergence_win must be greater than or equal to 2.

The default value of this parameter is elec_convergence_win.

2.108 integer :: optics_nbands

The number of bands at each k-point when performing an optical spectrum calculation.

If OPTICS_NBANDS is present in the parameter file, the value of bs_nbands is determined by this keyword.

If OPTICS_NEXTRA_BANDS is specified optics_nbands = max(nup, ndown) + OPTICS_NEXTRA_BANDS.

If OPTICS_PERC_EXTRA_BANDS is specified optics_nbands = $\mathtt{max}(\mathtt{nup}, \mathtt{ndown}) \times (1 + \frac{\mathrm{OPTICS_PERC_EXTRA_BANDS}}{100})$.

The default value of this parameter is $3 \times$ nbands.

2.109 character(len=15) :: optics_xc_functional

This parameter determines the exchange-correlation functional used for calculation of optical spectra. The possible values are the same as those for bs_xc_functional.

The default value of this parameter is bs_xf_functional.

2.110 character :: tssearch_method

The method used for transition state searches. Possible values for this parameter are:

- LSTQST

The default value of this parameter is 'LSTQST'.

2.111 character :: tssearch_lstqst_protocol

The protocol used for an LST/QST transition state search. Possible values of this parameter are:

- LSTMaximum
- Halgren-Lipscomb
- LST/Optimization
- CompleteLSTQST
- QST/Optimization

The default value of this parameter is 'LSTMAXIMUM'.

2.112 integer :: tssearch_qst_max_iter

The maximum number of QST iterations during an LST/QST transition state search.
This parameter must be greater than zero.
The default value of this parameter is 5.

2.113 integer :: tssearch_cg_max_iter

The maximum number of conjugate gradients steps during an LST/QST transition state search.
This parameter must be greater than zero.
The default value of this paramter is 20.

2.114 real :: tssearch_force_tol

The tolerance for the ionic forces to accept convergence during a transition state search. The maximum ionic force must be less than this value for the convergence criteria to have been met.
This parameter must be greater than zero.
The default value of this parameter is 0.005 Hartree/Bohr.

2.115 real :: tssearch_disp_tol

The tolerance for the ionic displacement to accept convergence during a transition state search. The maximum ionic displacement during the last iteration must be less than this value for the convergence criteria to have been met.
This parameter must be greater than zero.
The default value of this parameter is 0.01 Bohr.

2.116 integer :: phonon_max_cycles

The maximum number of cycles in the phonon linear response minimiser.
The default value of this parameter is 50.

2.117 integer :: phonon_max_cg_steps

The maximum number of conjugate gradient minimisation steps in one cycle of the phonon linear response minimiser.
The default value of this parameters is 20.

2.118 real :: phonon_energy_tol

The tolerance for accepting convergence of the second order energy in an phonon linear response minimisation.
The default value of this parameter is the value of `elec_energy_tol`.

2.119 integer :: phonon_convergence_win

The second order energy must lie within phonon_energy_tol for the last phonon_convergence_win iterations for the convergence criteria to be met in the phonon linear response minimiser.

The value of phonon_convergence_win must be greater than or equal to 2.

The default value of this parameter is 2.

2.120 character(len=8) :: phonon_preconditioner

The preconditioner to use in the phonon minimisation algorithm.

Allowed values are:

- TPA
- RTPA

The default value for this parameter is 'TPA'

2.121 logical :: phonon_use_kpoint_symmetry

If TRUE then, for each phonon q-vector, perform the linear response calculation using the irreducible k-point set of the (reduced) symmetry. If false, use the complete fully symmetric k-point set.

The default value of ths parameter is TRUE.

2.122 character(len=20) :: phonon_method

This determines the method used to calculate the elements of the dynamical matrix. Possible values are:

- FINITEDISPLACEMENT - A finite displacement method will be used to numerically approximate the dynamical matrix. N.B. This can only be used for qpoint=(0,0,0).
- LINEARRESPONSE - Linear response will be used to calculate the dynamical matrix. This can be applied for arbitraty qpoint.

The alias 'DFPT' is accepted for 'LINEARRESPONSE' for the value of the PHONON_METHOD keyword. The parameter value will be set to 'LINEARRESPONSE' in this case.

The default value of this parameter is 'LINEARRESPONSE'.

2.123 real(kind=dp) :: phonon_dos_spacing

The resolution of the density of states calculation for phonons.

The default value of this parameter is 10 cm^-1.

2.124 real(kind=dp) :: phonon_finite_disp

The amplitude of the ion perturbation used in a finite displacement phonon calculation.

The default for this parameter is 0.01 atomic units.

2.125 logical :: phonon_calc_lo_to_splitting

Flag to select whether to compute non-analytic contribution to dynamical matrix from long-ranged electric field effects responsible for LO/TO splitting. This requires calculation of the dielectric permittivity by efield linear-response and the Born effective charges.

The default value of this parameter is TRUE.

2.126 logical :: phonon_sum_rule

Selects whether to explicitly correct the dynamical matrix to enforce the acoustic q=0 phonon sum rule, i.e. that 3 modes have zero frequency at q=0.

The default value of this parameter is FALSE.

2.127 logical :: calculate_born_charges

Selects whether to compute Born effective charge tensors as part of a phonon or efield linear-response calculation.

The default value of this parameter is TRUE.

2.128 logical :: born_charge_sum_rule

Selects whether to explicitly correct the Born effective charge tensor to enforce the sum rule that effective charges sum to zero.

The default value of this parameters is FALSE.

2.129 integer :: efield_max_cycles

The maximum number of cycles in the efield linear response minimiser.

The default value of this parameter is 50.

2.130 integer :: efield_max_cg_steps

The maximum number of conjugate gradient minimisation steps in one cycle of the efield linear response minimiser.

The default value of this parameters is 20.

2.131 real(kind=dp)l :: efield_energy_tol

The tolerance for accepting convergence of the second order energy in an efield linear response minimisation.

The default value of this parameter is 1×10^{-8} Bohr^3.

2.132 integer :: efield_convergence_win

The second order energy must lie within efield_energy_tol for the last efield_convergence_win iterations for the convergence criteria to be met in the efield linear response minimiser.

The value of efield_convergence_win must be greater than or equal to 2.

The default value of this parameter is 2.

2.133 logical :: efield_calc_ion_permittivity

This logical value determins if the ionic contribution to the permitivity is calculated during an efield run.

The default value of this parameter is TRUE.

2.134 character(len=20) :: efield_ignore_molec_modes

This string indicates the number of modes to ignore (translational and rotational) when calculating the ionic contribution to the permittivity and polarizability.

Possible values are:

- CRYSTAL - The 3 lowest modes are ignored.
- MOLECULE - The lowest 6 modes are ignored.

- LINEAR_MOLECULE - The lowest 5 modes are ignored.

The default value of this parameter is "CRYSTAL".

2.135 Pseudopotentials

CASTEP supports several different pseudpotential file formats distinguished by a filename suffix. The name of the file should appear following the chemical symbol of the element in a `%BLOCK SPECIES_POT` in the `.cell` file.

`.usp` Vanderbilt ultrasoft pseudopotentials

`.usp` Vanderbilt ultrasoft pseudopotentials with a pseudocore charge density.

`.recpot` Norm conserving potential in the old CASTEP format.

`.DAT` TM potential files, generated by atm+kbconv (Toullier, Martins, Froyen ...)

If a real-space application of the pseudopotential projectors is required (`pspot_nonlocal_type = real` in *seed*`.param`) then the real-space projectors are read from a file named *seed*`..realpot`. This may be generated using Accelrys Inc's Cerius 2 or materials Studio software.

3 output files

Like the input files, all CASTEP 3.02 output files take the form

seedname.extn.

Mostly the output files are identical between serial and MPI parallel run cases. The exceptions are the wavefunction and diagnostic error files which have a separate copy for each MPI process with rank *nnnn* labelled `.wvfn.`*nnnn* and `.err.`*nnnn* respectively.

The major output files are

3.1 `.castep`

The main output file is called

seedname`.castep`.

and contains all major data describing a run, all significant results computed for the task selected and progress information. The amount of logging information included in this file is controlled by the parameter `IPRINT`.

3.2 `.err.`*nnnn*

Every MPI process writes a file named *seedname*`.err.`*nnnn* where *nnnn* is the MPI rank. Any fatal error messages which cause an abnormal termination of a run are written to these files. This is the first place to look if a run appears to have terminated prematurely.

3.3 `.check`

The `.check` file is a Fortran unformatted (ie binary) file written at the end of a successful single-point energy calculation and periodically throughout a geometry optimization, molecular dynamics run. It serves as a complete record of all information evaluated so far during a run, including the ground-state kohn-sham orbitals, the charge density on the FFT grid and any quantities including the fermi energy, forces, stresses, unit cell during optimization or MD.

The *seedname*`.check` file may be read by Accelrys Materials Sudio 3.0 or by the utilities in the `New_Code/Source/Tools` directories for analysis.

A run may also be continued from a *seedname*`.check` file in the case where a run which was interrupted, or it is desired to continue with a change of parameters. This is achieved by setting the parameter `CONTINUATION`.

3.4 `.wvfn.`*nnnn*

Upon a successful single-point energy calculation the wavefunctions are written to these files (one per MPI process). This is not the main restart mechanism, but it is possible to restart a run using these by setting the parameter `ELEC_RESTORE_FILE` to the root name (`i.e.` strip the processor rank). The location these are written to is set by parameter `ELEC_DUMP_FILE`.

3.5 `.bands`

The `.bands` file is written after a single-point energy calculation or bandstructure run and contains the electronic eigenvalues.

3.6 `.phonon`

A *seedname*`.phonon` file is written during a phonon linear response calculation and contains a header and the phonon frequencies and eigenvectors for every q-point computed.

3.7 `.geom`

A record of the atomic co-ordinates at each geometry optimization iteration are written to the file *seedname*`.geom`.

3.8 `.md`

A record of the atomic co-ordinates at each molecular dynamics iteration are written to the file *seedname*`.md`.

3.9 Wavefunction paging and Temporary files

CASTEP 3.02 may also use a number of Fortran `STATUS=SCRATCH` files during a run. In particular the strategy of *wavefunction paging* may be used. This means that the coefficients of a single band and k-point are retained in memory at any one time and the others are stored in the temporary disk file. This is selected by parameters `PAGE_WVFNS` and `OPT_STRATEGY=MEMORY`.

Most Unix or Linux Fortran compilers store these files in the current directory. Though hidden these files do use disk space, which may lead to mysterious exceeding of disk quota. In many cases this location may be changed by setting the environment variable `TMPDIR` to another directory path. The mechanism for changing temporary file directories should be documented in your compiler's manual.

4 Auxilliary tools

This release contains a number of programs and tools in addition to CASTEP. Some of these are PERL scripts which require the Perl 5 language to be installed on your system. Others are Fortran and are compiled by the command "Make tools". Only sketchy documentation is available at this stage.

4.1 Fortran Tools

`charge2d` Extracts a 2d slice of charge density from the *seed*`.check` file for plotting. See file `example2d.in` in the `New_Code/Source/Tools` directory.

geom2xtl This program will convert a *seedname*`.geom` output file from a CASTEP 3.02 geometry optimisation run into standard `.xtl` format for viewing.

geom2xyz This program will convert a *seedname*`.geom` output file from a CASTEP 3.02 geometry optimisation run into standard `.xtl` format for viewing or animation.

mdxyz This program will convert a *seedname*`.md` output file from a CASTEP 3.02 geometry optimisation run into standard `.xtl` format for viewing or animation.

castep2casino The CASTEP2CASINO program takes a model file from a Castep calculation and writes out a Casino-formatted file in *seedname*`.casino` – this needs to be renamed pwfn.data in order for Casino to recognise it.

bands2dos This program will convert a ¡seedname¿.bands output file from a NEWTEP bandstructure run into a DOS(E) x-y format for plotting. NB Input file gives energies in AU (ie Hartrees) whereas we give an output file in eV ...

4.2 PERL Tools

dispersion.pl A PERL program for plotting phonon dispersions or band structures. (Requires perl 5.005 or newer and xmgrace).

In basic use `dispersion.pl` reads the phonon frequencies from a *seedname*`.castep` or *seedname*`.phonon` file and prints out a table. If the option `-xg` is also given and the `xmgrace` graph-potting program is installed then the script will attempt to create a plot of the dispersion curves.

If the *seedname*`.phonon` file is given the script also makes a best attempt to determine where bands cross based on the eigenvectors read from the file. This may be tuned using the `-ftol` option which gives the maximum Δf considered when searching for branches to join.

The script can also read in a file of experimental data points and overplot these. Each line must contain 3 numbers specifying the $q-$point and one or more frequency data.

$$\begin{array}{cccccc} q_{1a} & q_{1b} & q_{1c} & f_{11} & f_{21} & \cdots \\ q_{2a} & q_{2b} & q_{2c} & f_{21} & f_{22} & \cdots \end{array}$$

If the file contains a blank line the data following is considered to belong to a separate dataset and will be plotted using a distinct symbol.

Alternatively it can plot bandstructures from a *seedname*`.castep` or *seedname*`.bands` file if the `-bs` option is given.

-xg Write a script and invoke GRACE to plot data.

-ps Invoke GRACE to plot data and write as a PostScript file.

-eps Invoke GRACE to plot data and write as an encapsulated. PostScript (EPS) file.

-np Do not plot data, write a GRACE script.

-bs Read band-structure from ¡¿.castep or ¡¿.bands.

-expt FILE Read experimental data from EXPT and overplot.

-dat Reread standard output from previous run and plot.

-ftol f Set maximum discrepancy tolerance for phonon branch joining.

-v Be verbose about progress

cteprouts The directory **cteprouts** contains a number of PERL programs for interconverting file formats. The calls usually take the form, *e.g.*

```
xtl2cell seed.xtl > seed.cell
```

converts a crystal structure file in the `XTL` format into a CASTEP 3.02 *seed*`.cell` file. The several scripts `newtep2`*xxx* convert CASTEP 3.02 `.castep` output files into various molecular graphics file formats.

Among the useful programs are `newtep2cell` which reads a CASTEP output file, eg from a geometry optimization and writes the final structure to a *seed*`.cell` file to begin a new run. `newgeom2dcd`, `newgeom2xyz` and `newgeom2pdbseq` convert the *seed*`.geom` file from a geometry optimization or MD to a standard trajectory format for animation.

See the file `New_Code/Source/Tools/cteprouts/README` for a complete list and more instructions.

5 Migration from CASTEP 4.2

The input files of CASTEP 3.02 have a completely different format from those of CASTEP 4.2. However CASTEP 3.02 has all of the capabilities of CASTEP 4.2 and can perform compatible calculations. These should give almost identical energies provided that the same pseudopotentials are used. The reason for the qualifier "almost" is that CASTEP 3.02 used the latest CODATA 97 values for the fundamental constants consistently throughout the program. Geometry optimization calculations using CASTEP3.02 will often produce slightly different results because and will be able to converge where CASTEP 4.2 did not, or converged poorly.

There are a number of utilities in the directory `New_Code/Source/Tools/cteprouts` which can be used to migrate a calculation. See section 4 for more information. In particular

1. The command

   `geom2cell` *`oldseed`*`.geom > ` *`newseed`*`.cell`

   will convert a CASTEP4.2 `.geom` file into a CASTEP 3.02 `.cell` file. Because the old `.geom` format contained no information on atom types you must supply these using an environment variable `ATOMS`. For example

   ```
   export ATOMS=Si:C:H
   geom2cell oldseed.geom > newseed.cell
   ```

   indicates that the 3 atoms in *oldseed*`.geom` are Si, C and H in that order.

2. The command

   `cst2cell` *`oldseed`*`.cst > ` *`newseed`*`.cell`

   will convert a CASTEP4.2 `.cst` output file into a CASTEP 3.02 `.cell` file. This is able to extract more information, than `geom2cell` including atom types.

In both cases the output files will probably not be exactly what is required and will need some minor changes, particularly specification of pseudopotentials.

CASTEP 4.2 read its pseudopotentials from a single large file, which was the concatenation of the individual files for all of the elements. In contrast CASTEP 3.02 requires that the pseudpotental file are specified by name in a `block species_pot` in the `.cell` file. A comprehensive selection of pseudopotential files is supplied in the database.

References

[1] R. Car and M. Parrinello, *Unified approach for molecular-dynamics and density-functional theory*, Phys. Rev. Lett. **55** (1985), no. 22, 2471–2474.

[2] Lyndon J. Clarke, Ivan Štich, and Mike C. Payne, *Large-scale ab initio total energy calculations on parallel computers*, Comp. Phys. Commun. **72** (1992), no. 4, 14–28.

[3] X. Gonze and C. Lee, *Dynamical matrices, born effective charges, dielectric permittivity tensors, and interatomic force constants from density-functional perturbation theory*, Phys. Review B **55** (1997), 10355–10368.

[4] K. Laasonen, R. Car, C. Lee, and D. Vanderbilt, *Implementation of ultrasoft pseudopotentials in abinitio molecular-dynamics*, Phys. Review B **43** (1991), 6796–6799.

[5] Richard M. Martin, *Electronic structure: Basic theory and methods*, Cambridge, 2004, ISBN: 0521782856.

[6] M. C. Payne, M. P. Teter, D. C. Allen, T. A. Arias, and J. D. Joannopoulos, *Iterative minimization techniques for* ab initio *total-energy calculations: molecular dynamics and conjugate gradients*, Rev. Mod. Phys. **64** (1992), no. 4, 1045–1097.

[7] M. D. Segall, P. J. D. Lindan, M. J. Probert, C. J. Pickard, P. J. Hasnip, S. J. Clark, and M. C. Payne, *First-principles simulation: ideas, illustrations and the castep code*, J. Phys.: Cond. Mat. **14** (2002), 2717–2744.

[8] D. Vanderbilt, *Soft self-consistent pseudopotentials in a generalized eigenvalue formalism*, Phys. Review B **41** (1990), 7892–7895.